1 Stabilizer Formalism [1]

In this section, we describe the stabilizer formalism, which forms the backbone of quantum error correction theory. Before delving into the main discussion, we provide a small case example of error correction and briefly introduce classical error correction theory.

1.1 Introduction to Error Correction

Noise is a great bane of information systems. Whenever we build systems for computation or other purposes, we cannot avoid noise from outside the system. However, even in such environments, classical components perform reliable computation with a failure rate typically below one error in 10^{17} operations. The details of the techniques used to protect against noise in practice are sometimes rather complicated, but the basic principles are easily understood. For example, consider the case when Alice sends Bob a single bit of information through a noisy channel, where the error rate is p. If Alice sends a bit without protection, a bit-flip error, where 0 flips to 1 and vice versa, occurs with a probability of p. As a result, Bob will fail to receive the correct information from Alice with a probability of p. In such cases, we cannot communicate reliably through the noisy channel. Yet, there is a way to address this issue. If Alice introduces redundancy for a bit of information, a process called "encoding," such as:

$$0 \to 000 \quad \text{or} \quad 1 \to 111, \tag{1}$$

Then Alice uses 2 additional bits to send a single bit of information, effectively using two additional channels. In this case, assume Alice sends 000. Each bit can be flipped with a probability of p. Bob then receives:

- No bit flipped, e.g., 000, with a probability of $(1-p)^3$,
- One bit flipped, e.g., 100, 010, 001, with a probability of $3p(1-p)^2$,
- Two bits flipped, e.g., 110, 101, 011, with a probability of $3p^2(1-p)$,
- Three bits flipped, e.g., 111, with a probability of p^3.

Thus, Bob will receive 000, 100, 010, or 001 with a probability of $(1-p)^3 + 3p(1-p)^2$. He then performs a majority vote between 0 and 1, allowing him to restore the information sent by Alice. For reliable communication through encoding, the probability of failure, $3p^2(1-p)+p^3$, must be less than $1/2$. This gives the threshold for reliable communication as $p < 1/2$. Encoding a bit of information into several bits by repeating the original is called a repetition code.

1.2 Error Correction Beyond Classical Ones

In quantum environments, we cannot perform error correction in the same way as described in Section 1.1, because performing a majority vote on an encoded qubit destroys the qubit state, collapsing it into one of the eigenstates of the observable. This is crucial, as in quantum computation, we cannot ignore noise while performing operations. We need to correct errors and perform computational operations simultaneously. If we destroy the states of qubits, we cannot continue the remaining computation.

There are also important differences between classical information and quantum information, requiring new ideas to make quantum error-correcting codes possible. In particular, at first glance, we encounter three rather formidable difficulties to address [1]:

- *No cloning*: One might try to implement the repetition code quantum mechanically by duplicating the quantum state three or more times. This is forbidden by the no-cloning theorem.
- *Errors are continuous*: A continuum of different errors may occur on a single qubit. Determining which error occurred in order to correct it would appear to require infinite precision, and therefore infinite resources.
- *Measurement destroys quantum information*: In classical error-correction we observe the output from the channel, and decide what decoding procedure to adopt. Observation in quantum mechanics generally destroys the quantum state under observation, and makes recovery impossible.

Fortunately, these difficulties are not fatal, and we will demonstrate a quantum version of the repetition code to illustrate this. In quantum error correction, there exist two types of errors, in contrast to classical computation: bit-flip errors and phase-flip errors. Bit-flip errors are the same as those in classical computation; for example, they are represented as $X|\psi\rangle$. Phase-flip errors occur when some qubits are acted on by Z, such as $Z|\psi\rangle$. from dualiality of X and Z, we only consider the case that bit-flip errors, and the case of phase-flip errors we can apply the same way as bit-flip. In this demonstration, we will demonstrate the repetition code for bit-flip errors. Suppose we encode the state $|\phi\rangle = a|0\rangle + b|1\rangle$ into a three-qubit space as $|\psi\rangle = a|000\rangle + b|111\rangle$. A useful way to express this is:

$$|0\rangle_L \to |000\rangle\,, \quad |1\rangle_L \to |111\rangle\,. \tag{2}$$

Here, $|\cdot\rangle_L$ is called the "logical qubit," while $|\cdot\rangle$ is called the "physical qubit." In this encoding scheme, we now obtain the encoding circuit shown below in Fig. 1.

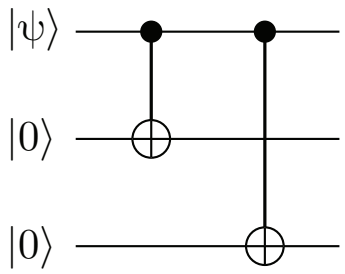

Fig. 1

Using the encoded state, we can obtain syndromes that indicate where an error has occurred. In this code, these syndromes can be determined by performing Pauli measurements of Z_1Z_2 and Z_2Z_3. For example, when no error has occurred on $|\psi\rangle$, we will obtain syndromes of 1 and 1 from Z_1Z_2 and Z_2Z_3, respectively, due to a straightforward calculation:

$$\begin{aligned} Z_1Z_2|\psi\rangle &= aZ_1Z_2|000\rangle + bZ_1Z_2|111\rangle \\ &= a|000\rangle + b|111\rangle \\ &= |\psi\rangle \end{aligned}$$

$$\begin{aligned} Z_2 Z_3 \left|\psi\right\rangle &= a Z_2 Z_3 \left|000\right\rangle + b Z_2 Z_3 \left|111\right\rangle \\ &= a \left|000\right\rangle + b \left|111\right\rangle \\ &= \left|\psi\right\rangle \end{aligned}$$

However, if an error has occurred on the first qubit of the three, we will obtain syndromes of -1 and 1 from $Z_1 Z_2$ and $Z_2 Z_3$, respectively, due to a straightforward calculation:

$$\begin{aligned} Z_1 Z_2 \left|\psi\right\rangle &= a Z_1 Z_2 \left|100\right\rangle + b Z_1 Z_2 \left|011\right\rangle \\ &= -a \left|100\right\rangle - b \left|011\right\rangle \\ &= - \left|\psi\right\rangle \end{aligned}$$

$$\begin{aligned} Z_2 Z_3 \left|\psi\right\rangle &= a Z_2 Z_3 \left|100\right\rangle + b Z_2 Z_3 \left|011\right\rangle \\ &= a \left|100\right\rangle + b \left|011\right\rangle \\ &= \left|\psi\right\rangle . \end{aligned}$$

Thus, we can determine that a bit-flip error has occurred on the first qubit and correct the error. In this case, we can correct a single error on any qubit. However, we cannot correct two or three errors. For instance, in the case of two errors on the second and third qubits of the three, we will obtain the following syndromes:

$$\begin{aligned} Z_1 Z_2 \left|\psi\right\rangle &= a Z_1 Z_2 \left|011\right\rangle + b Z_1 Z_2 \left|100\right\rangle \\ &= -a \left|011\right\rangle - b \left|100\right\rangle \\ &= - \left|\psi\right\rangle \end{aligned}$$

$$\begin{aligned} Z_2 Z_3 \left|\psi\right\rangle &= a Z_2 Z_3 \left|011\right\rangle + b Z_2 Z_3 \left|100\right\rangle \\ &= a \left|110\right\rangle + b \left|100\right\rangle \\ &= \left|\psi\right\rangle . \end{aligned}$$

These syndromes are the same as in the former case, so we cannot determine whether a single error has occurred on the first qubit or two errors have occurred on the second and third qubits. in the case of three errors, we will obtain the following syndromes:

$$\begin{aligned} Z_1 Z_2 \left|\psi\right\rangle &= a Z_1 Z_2 \left|111\right\rangle + b Z_1 Z_2 \left|000\right\rangle \\ &= a \left|111\right\rangle + b \left|000\right\rangle \\ &= \left|\psi\right\rangle \end{aligned}$$

$$\begin{aligned} Z_2 Z_3 \left|\psi\right\rangle &= a Z_2 Z_3 \left|111\right\rangle + b Z_2 Z_3 \left|000\right\rangle \\ &= a \left|111\right\rangle + b \left|000\right\rangle \\ &= \left|\psi\right\rangle , \end{aligned}$$

which makes it appear as though no errors have occurred, even though errors have actually occurred. In summary, this code can detect and correct one error, can detect two errors but cannot correct them, and cannot detect or correct three errors.

One may find that if a continuous error, such as $|0\rangle \rightarrow |0\rangle + |1\rangle$, occurs, we cannot obtain syndromes. Actually, when we obtain syndromes, we perform syndrome measurements using Z_1Z_2 and Z_2Z_3. As a result, the superposition state will be projected to either $|0\rangle$ or $|1\rangle$ with a probability proportional to the coefficients of the superposition state.

In the following section, we refer to the minimum number of errors that cannot be detected as the code distance, often expressed as d, encoded qubits as logical qubits, and the number of them as k, and qubits used to encode logical qubits as physical qubits, with their number often expressed as n. The properties of the code are often denoted as $[[n, k, d]]$. In the case of the former repetition code, its properties are $[[3, 1, 1]]$, but this code cannot correct phase-flip errors. Therefore, we show how to correct both types of errors: bit-flip and phase-flip.

1.3 Stabilizer

Stabilizer formalism is the most fundamental concept in quantum error correction. The stabilizer group is denoted as $\mathcal{S}$, and its elements are Pauli matrices, so $\mathcal{S} \subseteq \mathcal{P}$, but $-I \notin \mathcal{S}$. We also define the centralizer $\mathcal{C}(\mathcal{S})$, whose elements are Pauli matrices that commute with all elements of $\mathcal{S}$. Unlike classical codes, quantum codes are designed for computation while correcting errors, so there exist logical operators that change the states of logical qubits but leave stabilizer states unchanged. These logical operators L exist in $\mathcal{C}(\mathcal{S})\backslash\mathcal{S}$. In the context of Section 1.2, $\mathcal{S} = \{Z_1Z_2, Z_2Z_3\}$ and $\mathcal{C}(\mathcal{S}) = \{Z_1, Z_2, Z_3, X_1X_2X_3\}$, so in this case, the logical operators are $L_X = X_1X_2X_3$ and $L_Z = Z_1$, Z_2, or Z_3. As a side note, Z_1, Z_2, and Z_3 are equivalent logical operators in the quotient group of the stabilizer group.

In quantum error correction theory, there are representative codes, and we will provide an example of one, called the Steane Code [2]. The Steane Code consists of 7 qubits and 6 stabilizer generators, making it an $[[7, 1, 1]]$ code. Stabilizer generators form the basis of the stabilizer group, meaning all elements of the stabilizer group can be expressed as products of stabilizer generators. stabilizer generators and logical operators are shown in Tab. 1.

Table 1

Stabilizer Generators	Logical Operators
$I\ I\ I\ X\ X\ X\ X$	
$I\ X\ X\ I\ I\ X\ X$	
$X\ I\ X\ I\ X\ I\ X$	$L_X = X\ X\ X\ X\ X\ X\ X$
$I\ I\ I\ Z\ Z\ Z\ Z$	$L_Z = Z\ Z\ Z\ Z\ Z\ Z\ Z$
$I\ Z\ Z\ I\ I\ Z\ Z$	
$Z\ I\ Z\ I\ Z\ I\ Z$	

One may wonder how to encode a logical qubit in such codes. Encoding involves preparing a stabilizer state for the encoded qubits, so all you need to do is perform projective measurements of the stabilizer generators. First, you initialize all physical qubits to $|0\rangle$. This state is already a stabilizer state of the Z stabilizers listed in Tab. 1. Second, you only need to perform projective measurements of the X

stabilizers. Consequently, the encoded state $|0\rangle_L$ can be expressed as follows:

$$|0\rangle_L = \prod_i (I + s_i) |0000000\rangle$$

where s_i is a Z stabilizer generator, and the state is post-selected to the $+1$ eigenstates, ignoring the normalization factor. In the same way, $|1\rangle_L$ can be encoded. Thus, if you want to encode an arbitrary state $|\psi\rangle = a|0\rangle + b|1\rangle$, first prepare the Bell state $a|0000000\rangle + b|1111111\rangle$ and perform projective measurements of the Z stabilizer generators. The final states is:

$$|\psi_L\rangle = \prod_i (I + s_i) |0000000\rangle + \prod_i (I + s_i) |1111111\rangle .$$

From straightforward calculations, you can confirm that $L_X |0\rangle_L = |1\rangle_L$, $L_Z |1\rangle_L = -|1\rangle_L$, and $L_X L_Z = -L_Z L_X$. By definition, a logical operator cannot be expressed as the product of stabilizer generators.